Object-Oriented Design for Representation of Adaptive and Dynamic C2 Decision-Processing

Briefing to the DMSO Workshop on: The Representation of Command and Control Decision Making In Combat Simulations

27, 28 February 1996

William F. Waite
AEgis Research Corporation
6703 Odyssey Dr., Suite 200
Huntsville, AL 35806
phone: (205)922-0802
fax: (205)922-0904
email: Waite@AEgisRC.com

AEgis RESEARCH

OUTLINE

I. INTRODUCTION

II. INITIAL “THINKER’ CONSTRUCT

III. THINKER GENERALIZATION

IV. TRUTH vs. PERCEIVED DATA

V. ADOS PROTOTYPE

VI. CONCLUSION

AEgis RESEARCH

INTRODUCTION
- Thesis -

If... requisite design constraints on: 1) decision-processor class entities, 2) information interfaces, and 3) data management are applied;

then...the opportunities for flexible, powerful, dynamic, and adaptive representation of C2 decision processing are conceptually unlimited.

INTRODUCTION
- Exposition -

- Introduce a particular ‘thinker’ class construct
- Generalize the class concept
- Apply I/F and data management constraints
- Demonstrate efficacy of altrnative decision operator methods
- Provide net assessment

AEgis RESEARCH

OUTLINE

I. INTRODUCTION

II. INITIAL ‘THINKER’ CONSTRUCT
   A. Context
   B. Modeling Abstraction
   C. Thinker / ‘Ruleset’ Design

III. THINKER GENERALIZATION

IV. TRUTH vs. PERCEIVED DATA

V. ADOS PROTOTYPE

VI. CONCLUSION

INITIAL 'THINKER' CONSTRUCT
- Context -

- Extended Air Defense Test Bed (EADTB)
- Constructive Simulation
- Motivated by the need for an 'open' decision-processing representation
  - Defer decision processor component identification
  - Defer decision processing functionality
- Resulted in a 'point design'
  - Considerable flexibility and power of representation
  - User man-power intensive
  - Design constraints

INITIAL 'THINKER' CONSTRUCT
- Modeling Abstraction -

OBJECT PRINCIPAL CLASSES	METHODS (FUNCTIONAL MODELS)	
• Platform	- Moves (Guided Missile, Land-based, etc)	
	- State extrapolation	- Generate signature
	- Carry / Launch	- Service
	- Impair	- Assess damage
	- Domain of interest	- Volume intersection
• Sensor	- Scan	- Jam receive
	- Transmit	- Receive
	- Process	- Assess Damage
• Communications	- Device, antenna, link performance	
	- Path selection	- Path connectivity
	- ECCM effects	- Assess damage
• Thinker	- Decision processing	- Component command
• Environment	- Environmental and natural reference data	
	- Signal propagation and natural signatures	

OUTLINE

I. INTRODUCTION

II. INITIAL “THINKER’ CONSTRUCT

III. THINKER GENERALIZATION
   A. Context
   B. General Decision-Processor Class Object
   C. Application to C2 Representation

IV. TRUTH vs. PERCEIVED DATA

V. ADOS PROTOTYPE

VI. CONCLUSION

AEgis RESEARCH

THINKER GENERALIZATION
- Context -

- **Proposed to the Architecture Forum of the 9th DIS Workshop (Sep. '93)**
- **Suggested Decision-Processor (implicit) object class**
- **Suggested regularizing the partitioning of C2 decision-processing**
  - **Discriminate clearly communications message passing from simulation executive data passing**
  - **'Consistent 'wrappering' of C2 decision <u>operations</u> representation**
  - **Preservation of mapping to C2 decision-process <u>operator</u> representation**

THINKER GENERALIZATION
- Decision-Processor Class Object -

- **CONCEPT**
  - Single data-processing, decision-processing model class
  - Generates / receives Comms-to-Comms messages for communications model transactions
  - Processes sensor-perception and message content data
  - Issues commands to other local class components in composite entities
  - Contingency behaviors controlled by user-defined / data driven decision-operator 'methods'
- **BENEFITS**
  - Secure partition of 'truth' and 'perceived' data
  - Convenient encapsulation of decision-processing models
  - Consistent with MIL and decision SWIL designs

THINKER GENERALIZATION
- Application to C2 Representation-

- **C2 is ‘implicit’ in the behaviors of the decision-processor class entity components**
- **Command authority established as in the real-world by role assignment and acquiescence**
- **Behavior is contingent as desired (with ‘method’ TBD)**
- **Information interface (sensory and message data) is managed visibly**
- **Command and Control vs. Communications are clearly discriminated**

AEgis RESEARCH

OUTLINE

I. INTRODUCTION

II. INITIAL “THINKER’ CONSTRUCT

III. THINKER GENERALIZATION

IV. TRUTH vs. PERCEIVED DATA
   A. Context
   B. Definitions
   C. Data Domains

V. ADOS PROTOTYPE

VI. CONCLUSION

TRUTH vs. PERCEIVED DATA
- Context -

- Debated at the C4I SIG at the 11th and 12th DIS Workshops (March ‘95)
- Suggested formal acknowledgment of ‘truth’ and ‘perceived’ data domains
- Suggested formal guidance for data access by decision-processing entities
- Results were inconclusive

AEgis RESEARCH

TRUTH vs. PERCEIVED DATA
- Definitions -

'TRUTH DATA' - Data which is generated as part of the original, intrinsic, unequivocal representation (attribute values) of simulation domain conceptual entities, which constitute the one best value regarding the attributes of those entities as known to themselves and / or as manifest to the omniscient simulation executive.

Truth data includes that part of simulation representation domain data (entity state, tactical data message data, and environment data) which is generated by its originator and which is not subject to degradation.

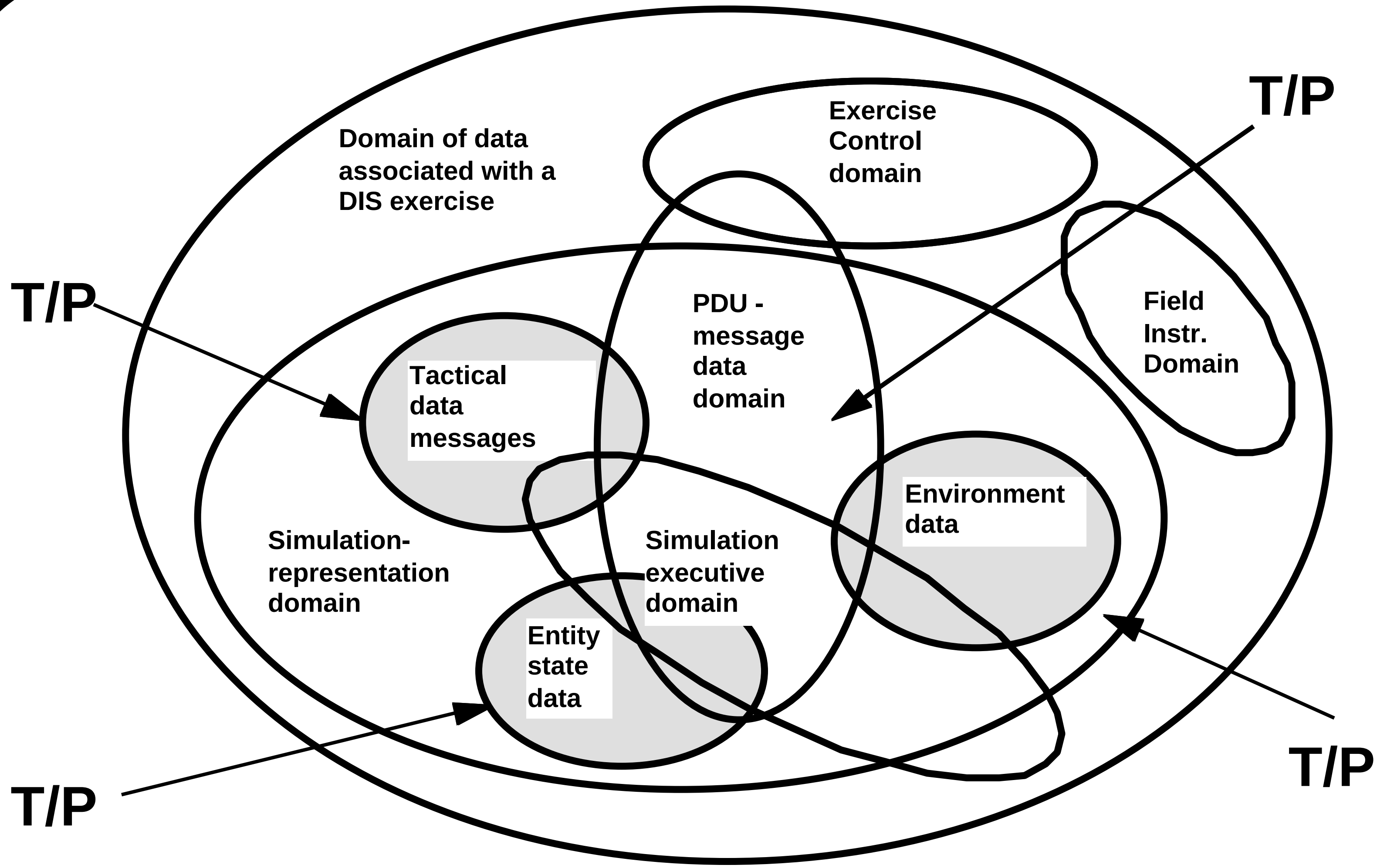
Truth is the *one* best value of what *is* in the exercise

TRUTH vs. PERCEIVED DATA - Definitions -

'PERCEIVED DATA' - Data which constitute the results of perception transformations (observation, derivation, estimation, or inference) by given entities of the truth attributes of other entities within the scope of the simulation / exercise and as manifest in (possibly corrupted, imprecise, inaccurate or otherwise imperfectly known) derived perceived-information data structures.

Perception is *derived* information which *might* be corrupted

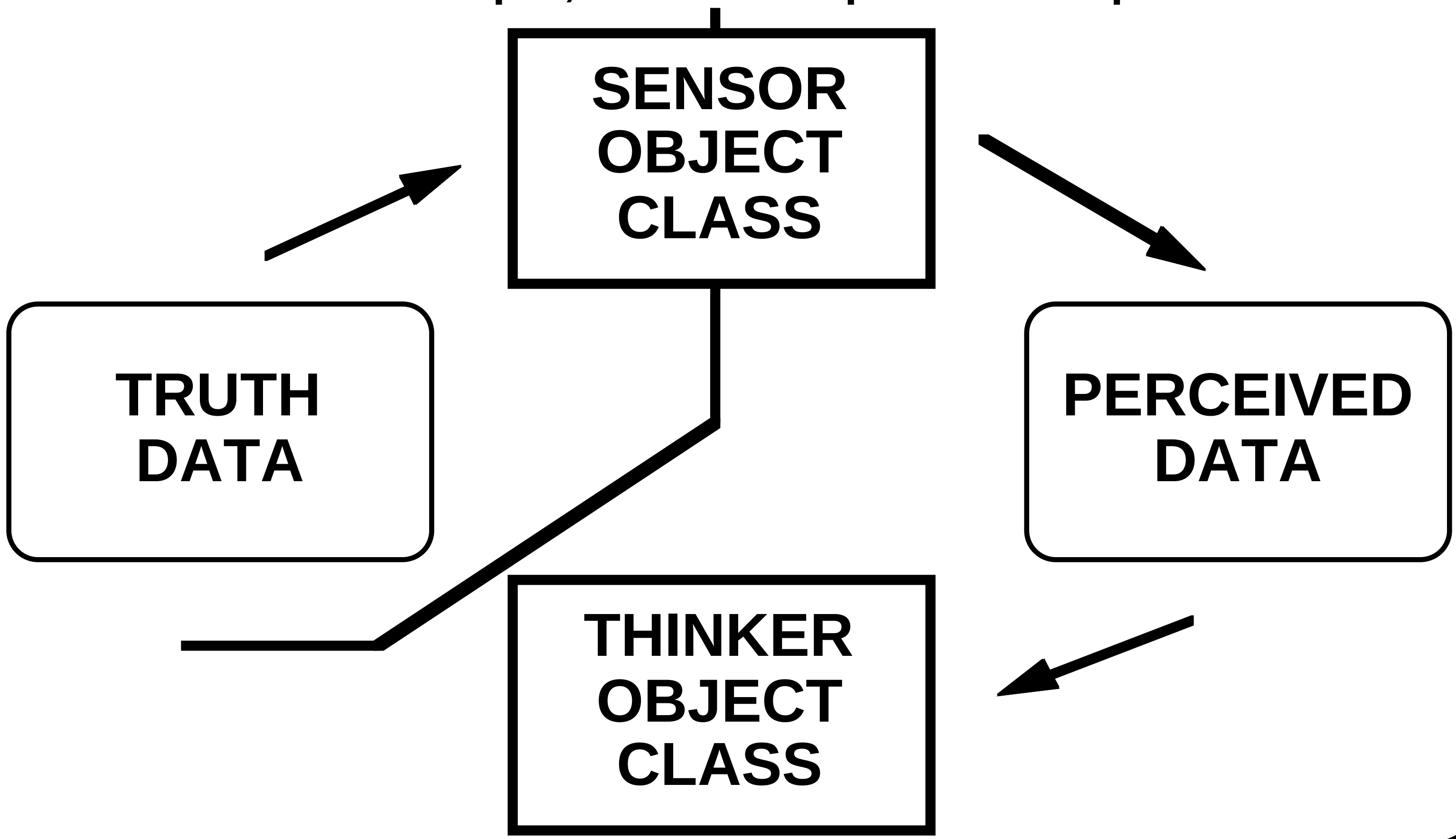
TRUTH vs. PERCEIVED DATA
- Definitions -



DATA DOMAIN DEFINITION

TRUTH vs. PERCEIVED DATA
- Definitions -

- **A symmetric 'dual' relationship pertains among truth and perceived *data* and sensor and thinker (decision processor) *objects***
  - **- Sensor objects transform data from truth to perceived domains**
  - **- Sensor has truth input, thinker has perceived input**



TRUTH vs. PERCEIVED DATA
- Proposed Guidance -

- Entity attribute values as originally derived are T-data
- Sensor input is T-data
- P-data should be able to have been corrupted
- Sensor output is P-data
- Received Tactical Data Messages are all P-data
- Input to C2 processing entities should be P-data
- The relationship between the cardinality of truth and perceived values of a data variable is 1 to 0-->n

AEgis RESEARCH

OUTLINE

I. INTRODUCTION

II. INITIAL “THINKER’ CONSTRUCT

III. THINKER GENERALIZATION

IV. TRUTH vs. PERCEIVED DATA

V. ADOS PROTOTYPE
   A. Context
   B. Design
   C. Lessons-Learned

VI. CONCLUSION

ADOS PROTOTYPE
- Context -

- **Conducted under the auspices of ESC / AVMW NASM prototyping**
  - Coordinated through JSIMS PO
  - Cooperation of USA MICOM
- **Constituted proof-of principle demonstration of generalized decision-processor object design**
  - Explicit object-oriented schema
  - Alternative decision processor operators
  - Dynamic / adaptive operations
- **Lessons-Learned documented**

ADOS PROTOTYPE
- Technical Issues -

The ADOS program is focused on the following issues:

- **Simulation Object-Oriented Design (OOD)**
  - **The design of a robust object schema**
  - **The construction of composite objects via aggregation vs. multiple inheritance**
  - **Processes by which real-world objects are mapped onto simulation objects / events**
  - ***The classification of decision processors***

- **Decision Processor Simulation**
  - ***The design of the decision processor class***
  - ***The segregation of ‘truth’ and ‘perceived’ data***
  - ***The ability of decision processors to learn***

AEgis RESEARCH

ADOS PROTOTYPE
- Technical Issues (cont'd) -

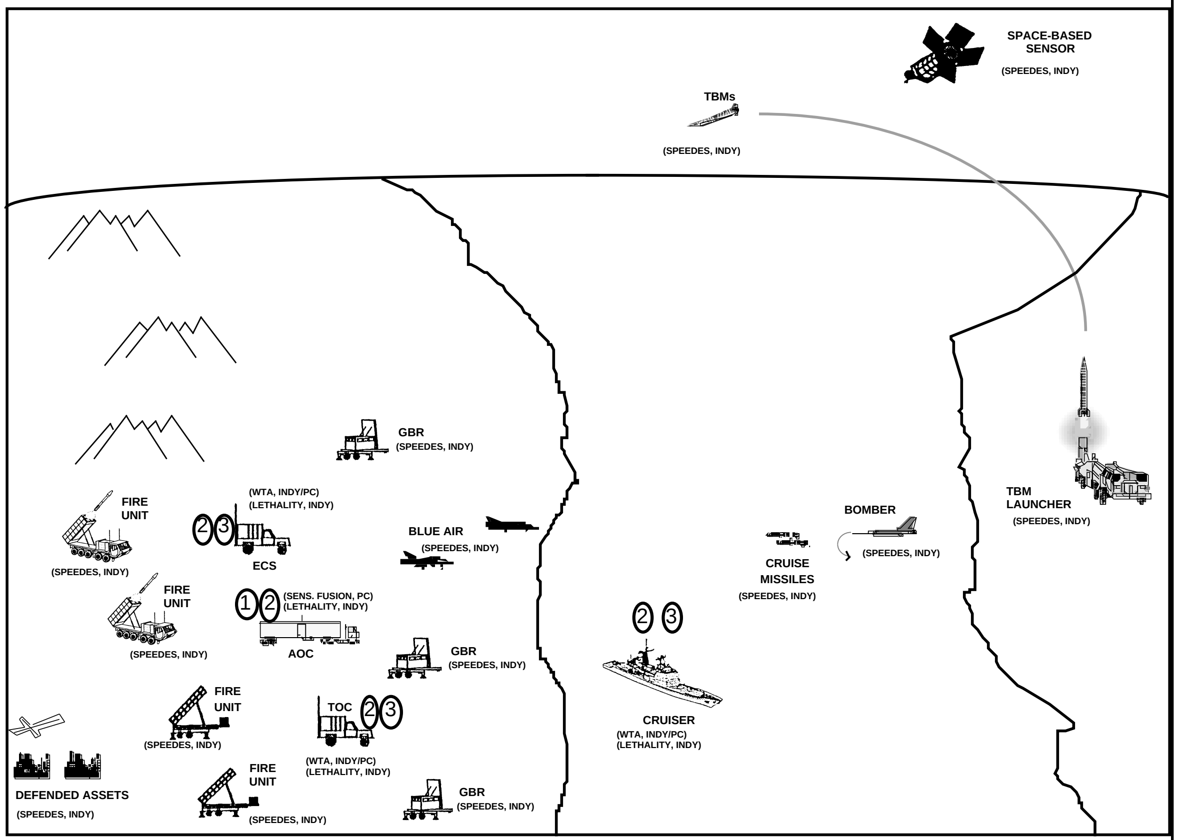
- **Distributed Discrete-Event Simulation**
  - ***The ability of decision processors (automated and MIL) to assume / relinquish control***
  - **Distributed platform processing**
  - ***Linkage to (live) forces, (virtual) simulators, and (constructive) software models***

- **Infrastructure Design**
  - **Infrastructure-to-model partition**
  - **General features**

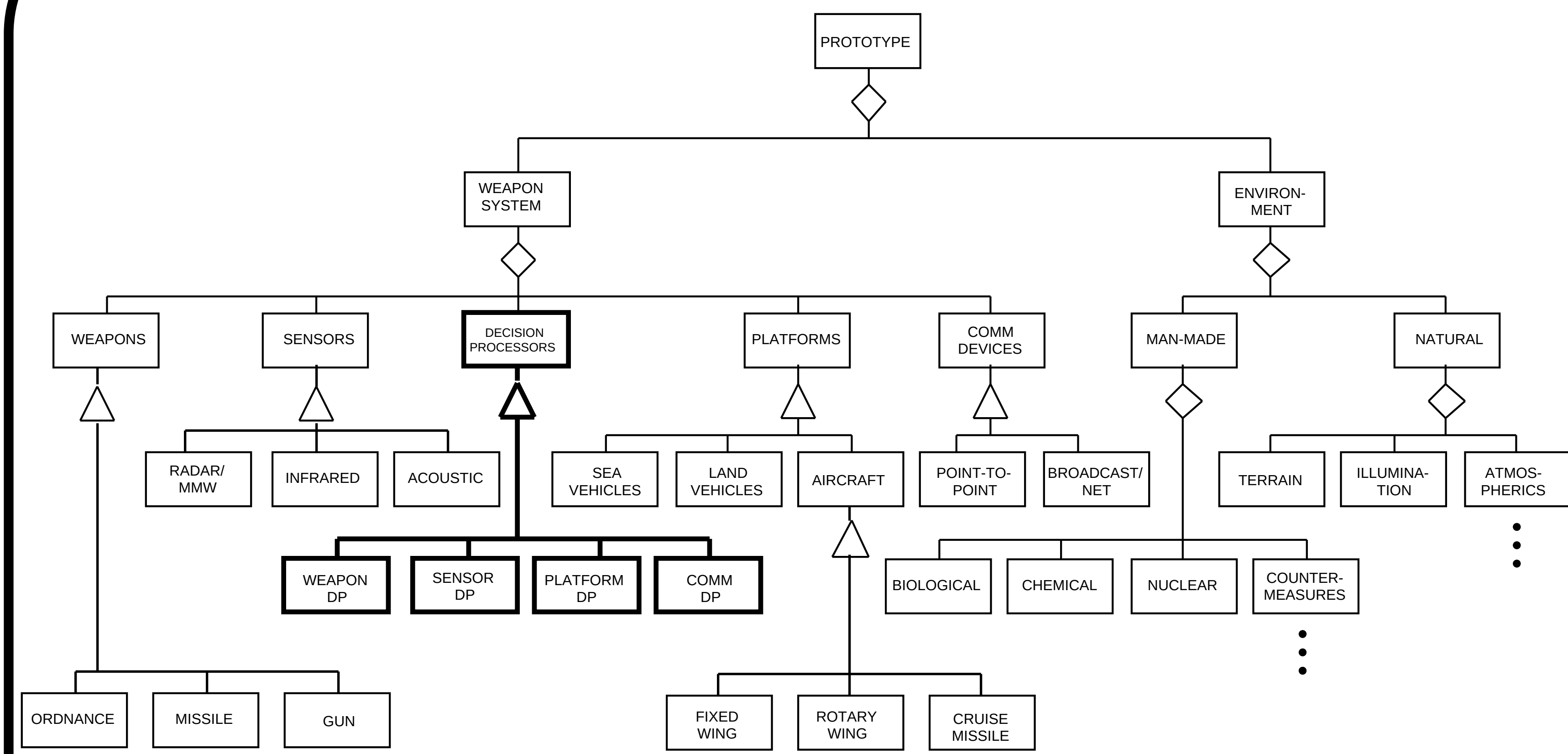
AEgis RESEARCH

ADOS PROTOTYPE
- Scenario-

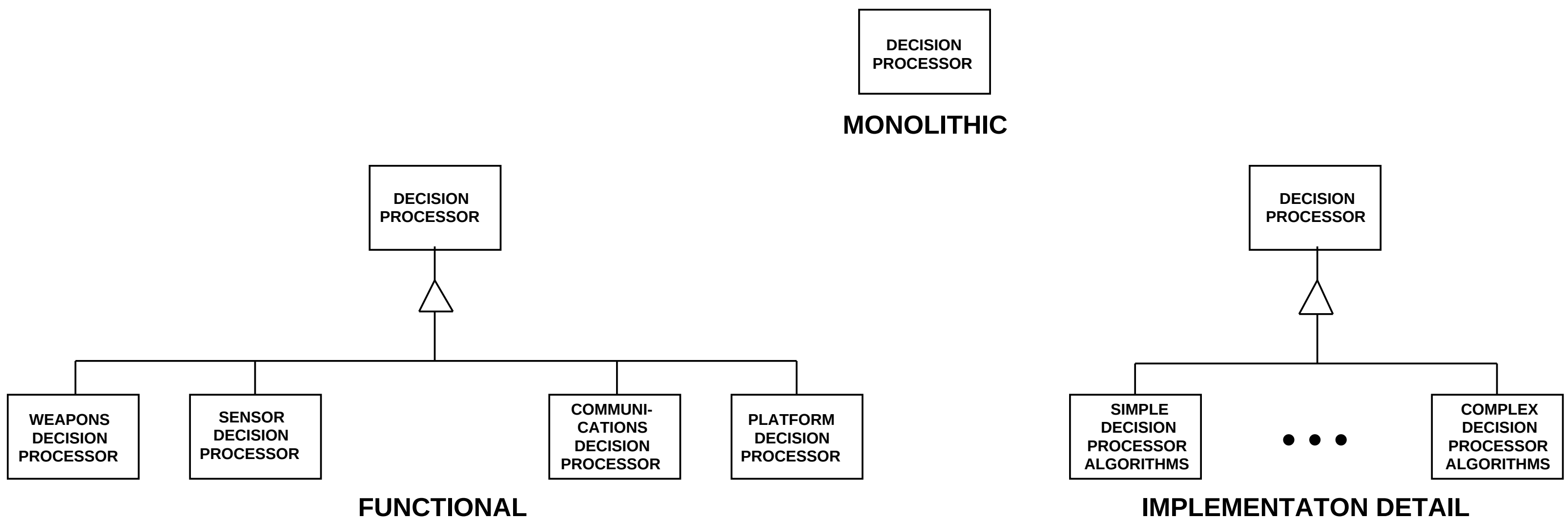
DECISION PROCESSORS	SCENARIO OBJECTS
1 SENSOR FUSION (C - PC)	WEAPON SYSTEMS SENSOR SYSTEMS C^3
2 DAMAGE PREDICTION (C - SUN SPARCSTATION)	C^3 WEAPON SYSTEMS DEFENDED ASSETS BATTLE MGMT
3 WEAPON-TGT ASSIGNMENT (C - SGI INDY) (EXPERT SYSTEM - PC) (FORTRAN - PC - MIL)	C^3 HOC^2 BATTLE MGMT WEAPON SYSTEMS FIRE UNITS



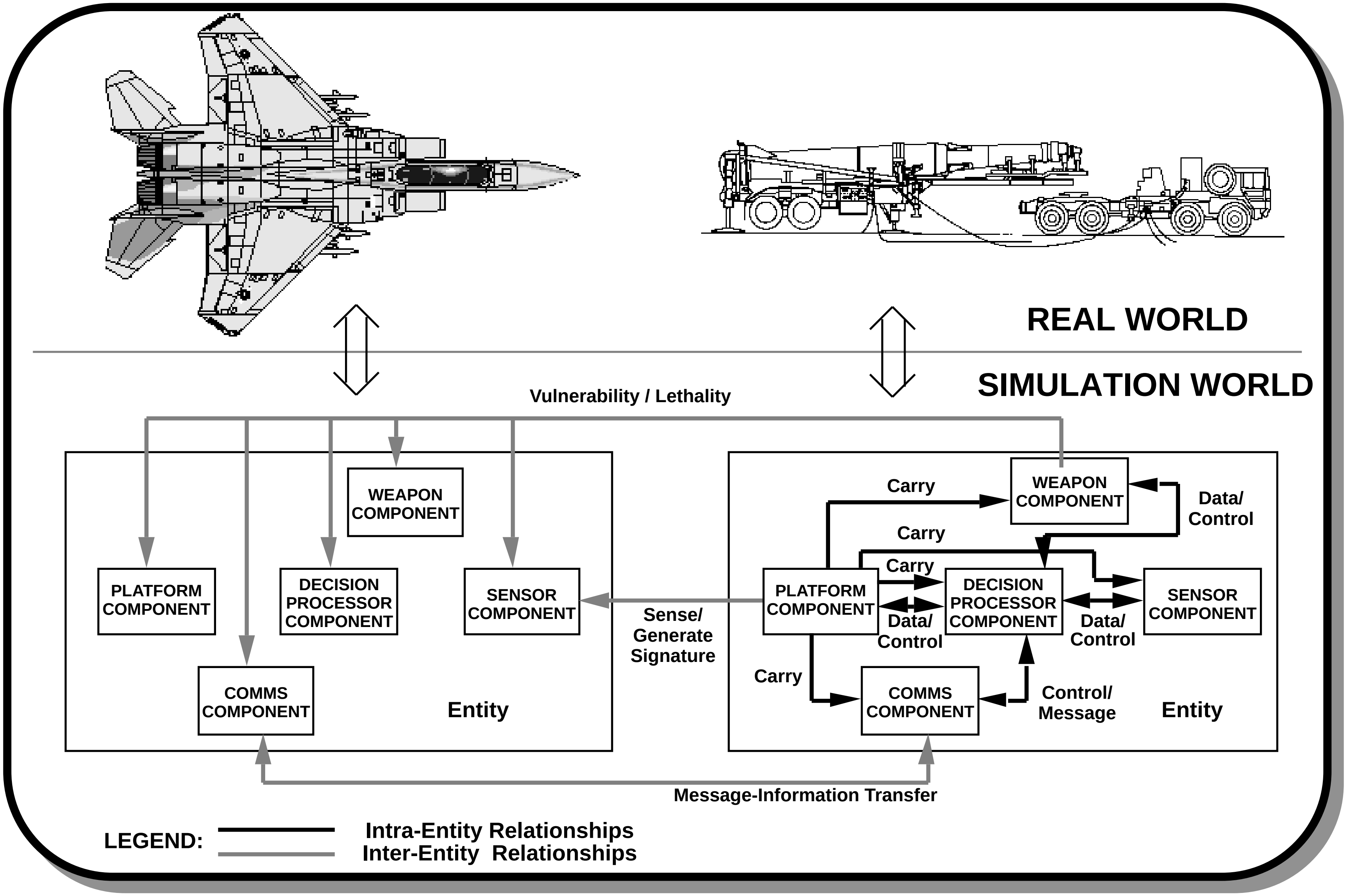
ADOS PROTOTYPE
- Object Class Specification -



ADOS PROTOTYPE
- Decision Processor Object Class -

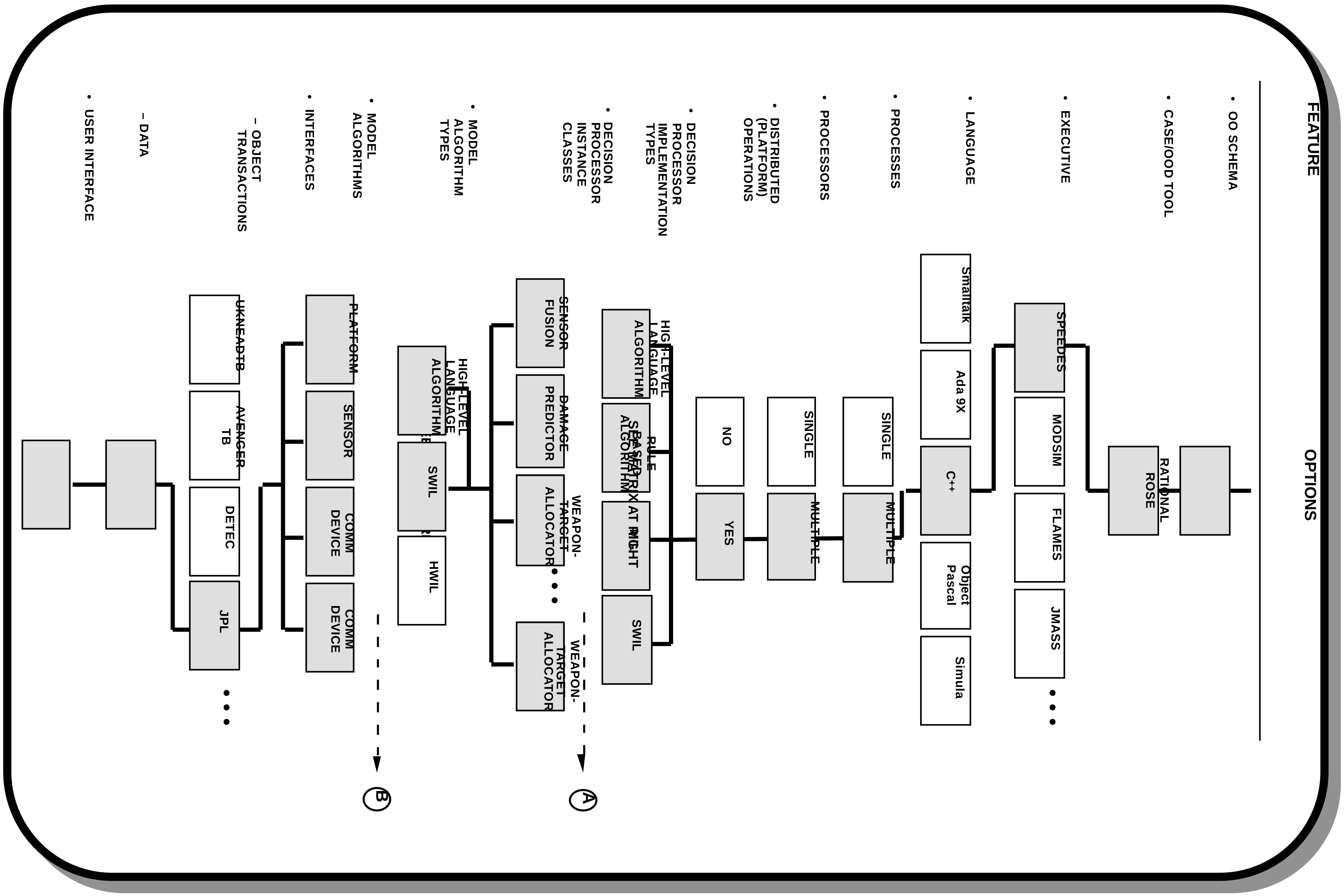


ADOS PROTOTYPE
- Object Interactions -



AEgis RESEARCH

ADOS PROTOTYPE
- Features / Options -



AEgis RESEARCH

ADOS PROTOTYPE
- Features / Options (cont'd) -

This matrix represents an initial concept of the implementation types which might be used to implement the various decision processor instance classes listed vertically.

(A)

DECISION PROCESSOR IMPLEMENTATION TYPES

DECISION PROCESSOR INSTANCE CLASSES	HIGH-LEVEL LANGUAGE ALGORITHM	RULE BASED ALGORITHM	MIL	SWIL
SENSOR FUSION				X
DAMAGE PREDICTOR	X			
WEAPON-TARGET ALLOCATOR	X		X	
WEAPON-TARGET ALLOCATOR		X		
WEAPON-TARGET ALLOCATOR				X

(B)

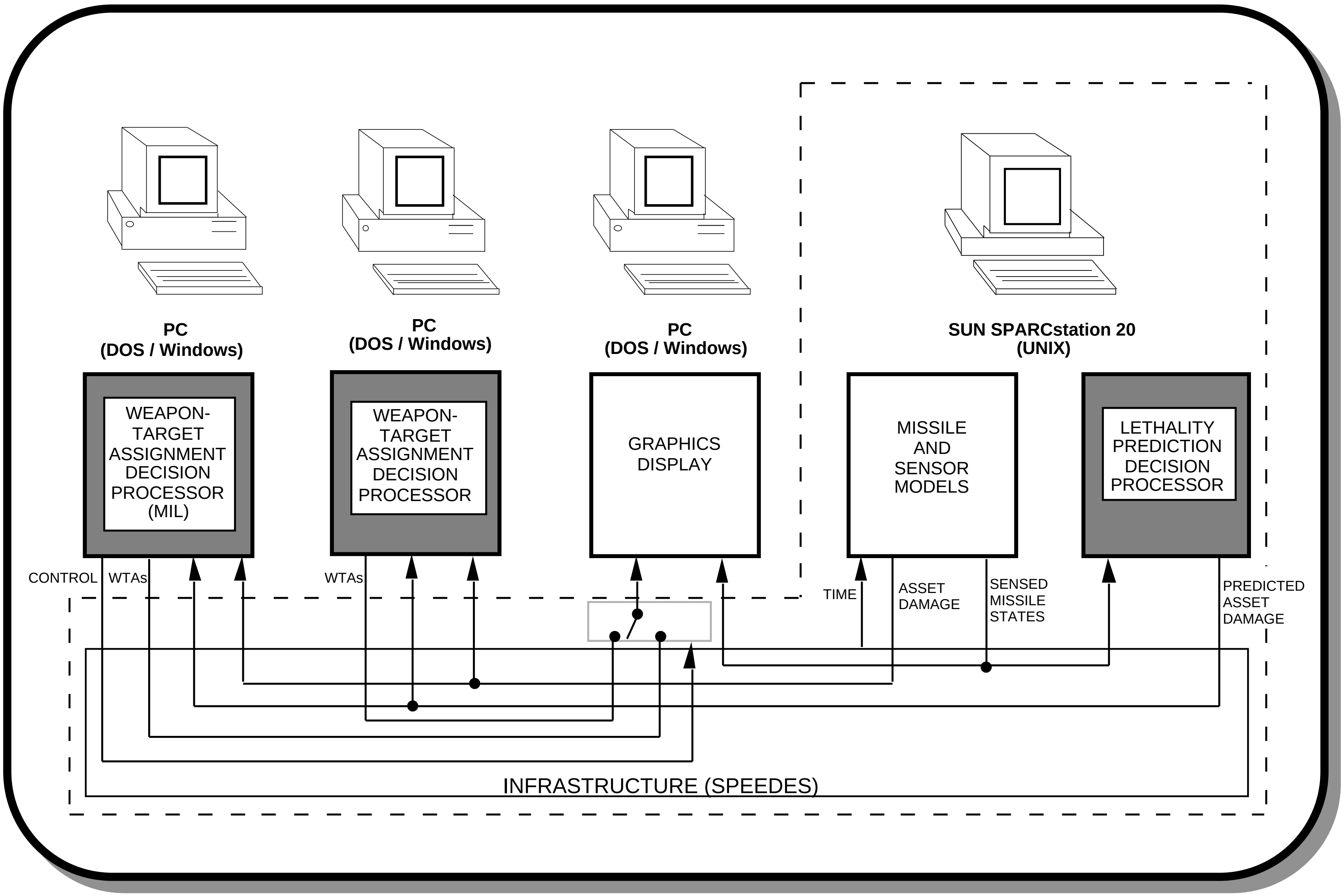
This matrix represents an initial concept of the model algorithm types which might be used to implement the various model algorithms listed vertically.

MODEL ALGORITHM TYPES

MODEL ALGORITHMS	HIGH-LEVEL LANGUAGE ALGORITHM	SWIL	HWIIL
PLATFORM	X		
SENSOR	X	(?)	
COMM DEVICE	X	(?)	
WEAPON	X	(?)	

AEgis RESEARCH

ADOS PROTOTYPE
- Architecture -



ADOS PROTOTYPE
- Decision Processors -

Weapon-Target Assignment MIL Model

• **Source:**	**AEgis Research Corporation**
• **Function:**	**Assigns weapons to incoming targets**
• **Implementation:**	**High-level language algorithm with MIL interface**
• **Platform:**	**PC**
• **Language:**	**FORTRAN**
• **Inputs:**	**Time and incoming target position and velocity** **Predicted damage data**
• **Outputs:**	**Weapon-target pairings** **Fire commands**
• **Comments:**	**Will switch in real-time with automated WTA**

ADOS PROTOTYPE
- Decision Processors -

Weapon-Target Assignment Model

• Source:	MICOM
• Function:	Assigns weapons to incoming targets
• Implementation:	High-level language algorithm
• Platform:	PC
• Language:	FORTRAN
• Inputs:	Time and incoming target position and velocity Predicted damage data
• Outputs:	Weapon-target pairings Fire commands
• Comments:	Will switch in real-time with MIL WTA

AEgis RESEARCH

ADOS PROTOTYPE
- Decision Processors (cont'd) -

Neural Net Damage Predictor

- **Source:** AEgis Research Corporation
- **Function:** Predicts damage by incoming targets
- **Implementation:** Neural Net
- **Platform:** Sun SPARCstation
- **Language:** C
- **Inputs:** Position and velocity of incoming targets
Dynamic asset values
- **Outputs:** Predicted damage values for each incoming target w.r.t. each defended asset
- **Comments:** Has the ability to learn

ADOS PROTOTYPE
- Lessons-Learned -

Simulation Object-Oriented Design (OOD)

- **The design of a robust object schema**

  -- An object schema is more than a class specification.
  -- Relationships are undervalued and under-specified.
  -- A parsimonious design is sufficient for representing much of the real world.
  -- It is neither convenient nor common to capture simulation representation domain and infrastructure implementation domain entities in a common object schema.
  -- Case tools are generally applicable, but require significant investment.
  -- Case tools enforce discipline and provide a self-consistent, persistent representation.

- **The construction of composite objects via aggregation vs. multiple inheritance**

  -- The use of composites is valuable.
  -- Either implementation will work.
  -- Multiple inheritance masks composition and is not allowed in some programming languages.

AEgis RESEARCH

ADOS PROTOTYPE
- Lessons-Learned (cont'd) -

Simulation Object-Oriented Design (OOD) (cont'd)

- **Processes by which real-world objects are mapped onto simulation objects / events**

  -- The simulation-domain object schema and the real world application-domain object schema should be mutually consistent.

  -- The most effective single constraint on data flow design is that simulation domain communications interfaces mimic real world interfaces. It is necessary to have an infrastructure that supports this.

- ***The classification of decision processors***

  -- The best object schema is one that maps to decision processor logic (methods) rather than to I/O data structures or applications.

ADOS PROTOTYPE
- Lessons-Learned (cont'd) -

Decision Processor Simulation

- ***The design of the decision processor class***

  -- The simulation architecture should be insensitive to the structure of the decision processor.
  -- Real world command and control interfaces should be used.

- ***The segregation of 'truth' and 'perceived' data***

  -- Implementing this idea is valuable, feasible, and inexpensive.
  -- This is easily abrogated by peculiarities of the simulation executive and related infrastructure design.
  -- An object schema which contains a decision processor class is valuable here.

ADOS PROTOTYPE
- Lessons-Learned (cont'd) -

Decision Processor Simulation (cont'd)

- ***The ability of decision processors to learn***

  -- Neural net training time may be large; a training environment is recommended.
  -- Neural net performance may degrade drastically as live conditions vary from training conditions.
  -- The application should drive the selection of the neural net class.

ADOS PROTOTYPE
- Lessons-Learned (cont'd) -

Distributed Discrete-Event Simulation

- *The ability of decision processors (automated and MIL) to assume / relinquish control*

    -- State initialization is problematic.
    -- This is partly a simulation-domain problem and partly an application-domain problem.
    -- Passive play "warm up" is recommended for MIL control assumption.

- Distributed platform processing

    -- "SPEEDES method" of interfacing distributed simulations yielded interesting challenges in several areas, e.g.: data formatting, differences between Unix sockets and Windows sockets, blocking techniques, C-to-FORTRAN interface, ...

    -- Communications problems among models must be solved, e.g. synchronization, time increment compatibility, etc.

ADOS PROTOTYPE
- Lessons-Learned (cont'd) -

Distributed Discrete-Event Simulation (cont'd)

- ***Linkage to (live) forces, (virtual) simulators, and (constructive) software models***

  -- Using an explicit, universal object schema and data interface is half the battle.

ADOS PROTOTYPE
- Lessons-Learned (cont'd) -

Infrastructure Design

- Infrastructure-to-model partition

  -- The infrastructure design should contain as few assumptions as possible about the simulation domain object schema.
  -- The infrastructure design should have practically no relationship to the simulation representation schema or to the application domain object schema (although the simulation representation schema and the application domain object schema should be highly coupled).

- General features

  -- The infrastructure should provide for requisite synchronization and preservation of causality, as appropriate.
  -- The infrastructure should provide all time management functions, including real-time and non-realtime control.
  -- The infrastructure should support alternative simulation executive types (time-step, discrete event, etc.)

OUTLINE

I. INTRODUCTION

II. INITIAL “THINKER’ CONSTRUCT

III. THINKER GENERALIZATION

IV. TRUTH vs. PERCEIVED DATA

V. ADOS PROTOTYPE

VI. CONCLUSION

NET ASSESSMENT

- **Implementing robust and flexible representation of C2 operators and operations is facilitated by:**
  - **Decision-processor class entity / component**
  - **Deliberate partition of truth and perceived data**
  - **Clearly defined information interfaces commensurate with real-world <u>sensor</u> and <u>tactical message</u> I/O**
- **Data management and data interface 'guidance' facilitates MIL and SWIL operations (C/V/L)**
- **With effective regularization of C2 object- and interface- definition, adaptive and flexible C2 simulation is limited only by imagination, implementation resources, and access to dynamic circumstantial data**